AFU

TRANSMITTAL OF APPEAL BRIEF (Large Entity)	Docket No. ITL.1083US

In Re Application Of: **Michael K. Eschmann**

Application No. 10/751,018	Filing Date December 31, 2003	Examiner Paul W. Schlie	Customer No. 21906	Group Art Unit 2186	Confirmation No. 3661

Invention: **Ordering Disk Cache Requests**

COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on
July 18, 2006

The fee for filing this Appeal Brief is: **$500.00**

☒   A check in the amount of the fee is enclosed.

☐   The Director has already been authorized to charge fees in this application to a Deposit Account.

☒   The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No.   **20-1504**

☐   Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Signature

Timothy N. Trop, Reg. No. 28,994
TROP, PRUNER & HU, P.C.
1616 S. Voss Road, Suite 750
Houston, TX 77057
713/468-8880 [Phone]
713/468-8883 [Fax]

Dated: **August 31, 2006**

CC:

P30LARGE/REV06

Applicant:		§
Michael K. Eschmann		§ Art Unit: 2186
		§
Serial No.: 10/751,018		§ Examiner: Paul W. Schlie
		§
Filed: December 31, 2003		§ Atty Docket: ITL.1083US
		§ (P18347)
For: Ordering Disk Cache Requests		§
		§ Assignee: Intel Corporation
		§

Mail Stop **Appeal Brief-Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
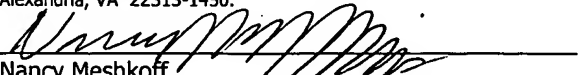
APPEAL BRIEF

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is the assignee Intel Corporation.

RELATED APPEALS AND INTERFERENCES

None.

STATUS OF CLAIMS

Claims 1-25 (Rejected).

Claims 1-25 are rejected and are the subject of this Appeal Brief.

STATUS OF AMENDMENTS

All amendments have been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

In the following discussion, the independent claims are read on one of many possible embodiments without limiting the claims:

1.   A method comprising:

determining if there is a pending demand request to a cached disk subsystem (Figure 3, diamond 56) and, if not, executing a non-demand request (Figure 3, diamond 58) (Specification at page 8, line 22-page 9, line 3).

FIG. 3
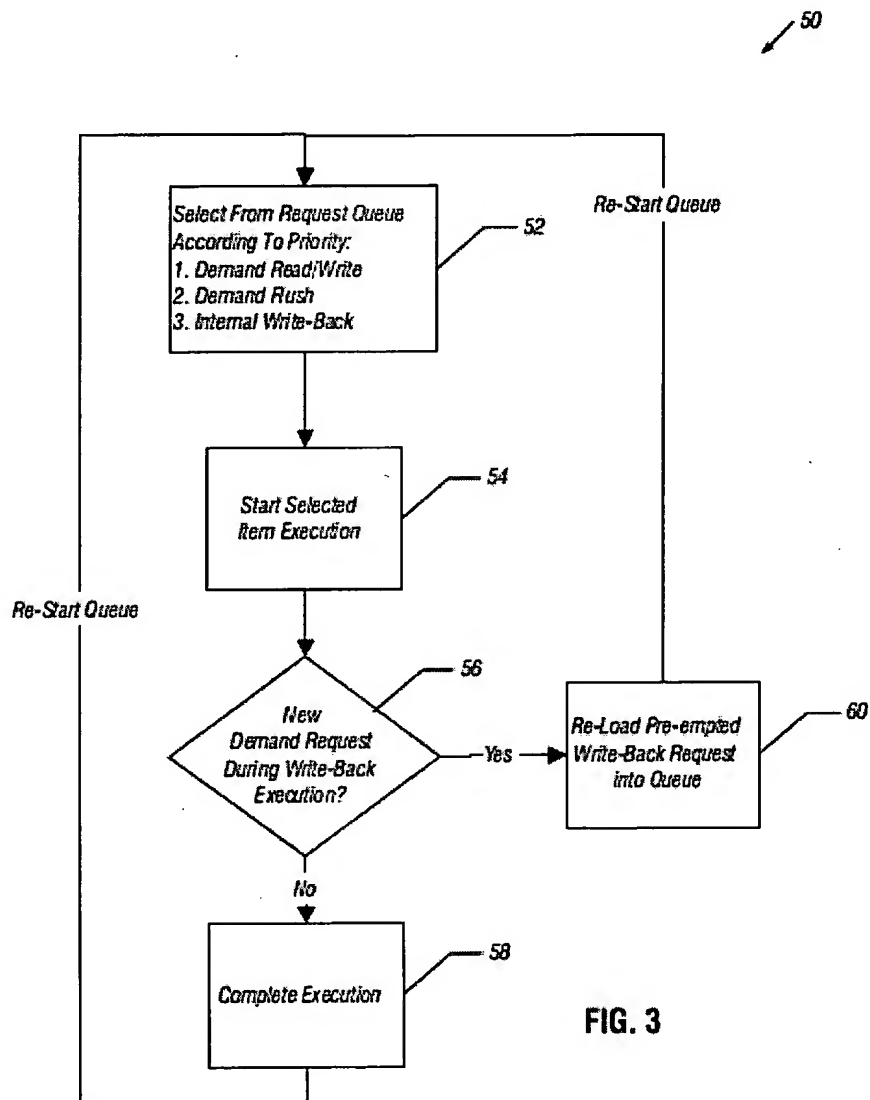
10.	An article comprising a medium storing instructions that, if executed, enable a processor-based system to:

determine if there is a pending demand request to a cached disk subsystem (Figure 3, diamond 56) and, if not, execute a non-demand request (Figure 3, diamond 58) (Specification at page 8, line 22-page 9, line 3).

18.	A system comprising:

a cache (Figure 1, 26);

a disk drive (Figure 1, 24) coupled to said cache; and

a controller (Figure 1, 20) to determine if there is a pending demand request to a cached disk subsystem and, if not, implement a non-demand request (Specification at page 8, line 22-page 9, line 3).
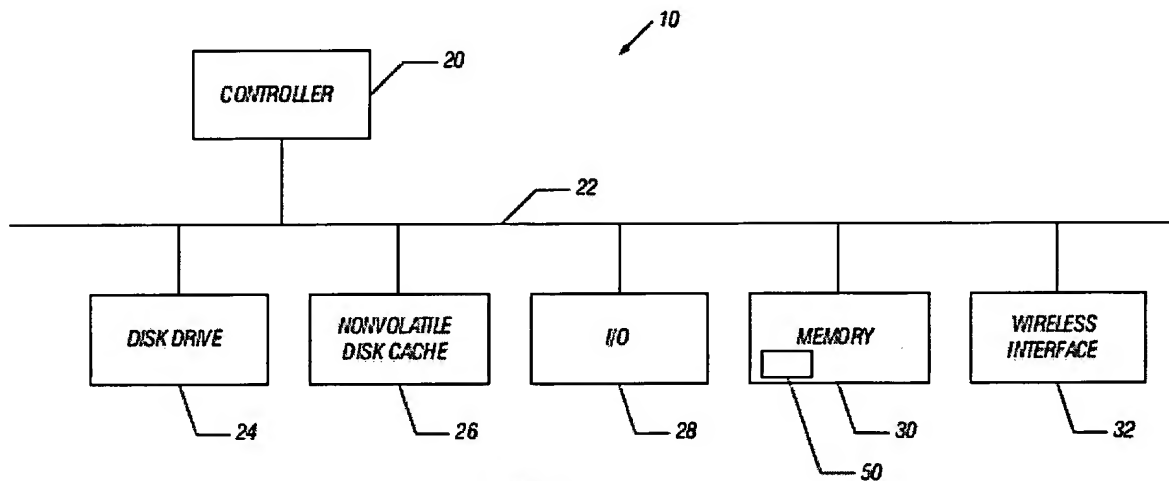
FIG. 1

At this point, no issue has been raised that would suggest that the words in the claims have any meaning other than their ordinary meanings. Nothing in this section should be taken as an indication that any claim term has a meaning other than its ordinary meaning.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A.   Are claims 1-25 anticipated under 35 U.S.C. § 102(e) or, in the alternative, rendered obvious under 35 U.S.C. § 103(a) by Herbst (U.S. Patent Publication No. 2003/0145165)?

ARGUMENT

A.   Are claims 1-25 anticipated under 35 U.S.C. § 102(e) or, in the alternative, rendered obvious under 35 U.S.C. § 103(a) by Herbst (U.S. Patent Publication No. 2003/0145165)?

Reversal is requested because there is no way that the cited reference could ever possibly meet the claimed invention, either by anticipation or by single-reference obviousness.

The claimed invention requires a prioritization scheme in which a certain type of request, called a demand request, is prioritized over a non-demand request. Demand requests may be, for example, a request to write data to or read data from a cache disk subsystem. If it is determined that there is a pending demand request to the cache disk subsystem, it is executed and then, and only then, or if not, executing non-demand requests. Thus the central concept is to determine whether there is a demand request, and only if there is not a demand request, execute a non-demand request. This requires that you always determine whether there are any demand requests.

The cited reference never distinguishes between demand requests and other requests. More pointedly, while not using the words "demand" or "non-demand", the cited reference does not distinguish between read or write requests and other type of request. Even more importantly, it never determines whether there is any type of demand request pending before executing non-demand requests.

The argument of inherency is not viable because the claim requires that you determine whether there is a demand request or not before you execute the non-demand request. The situation where there is only a non-demand request and it is executed does not fall within the claim. To execute a single pending request in the form of a non-demand request under the claimed invention requires, first, a check to determine whether there are any demand requests. This would never be done in the cited reference.

10

Applicant respectfully requests that each of the final rejections be reversed and that the claims subject to this Appeal be allowed to issue.

Respectfully submitted,

Date: <u>August 31, 2006</u>

Timothy N. Trop, Reg. No. 28,994
TROP, PRUNER & HU, P.C.
1616 S. Voss Road, Suite 750
Houston, TX 77057
713/468-8880 [Phone]
713/468-8883 [Fax]

Attorneys for Intel Corporation

CLAIMS APPENDIX

The claims on appeal are:

1.   A method comprising:

determining if there is a pending demand request to a cached disk subsystem and, if not, executing a non-demand request.

2.   The method of claim 1 including queuing requests including demand requests, requests to write from the cache back to a disk drive, and requests to flush the cache.

3.   The method of claim 2 wherein if the next request is a non-demand request, executing said non-demand request and monitoring for a demand request.

4.   The method of claim 3 including preempting the execution of the non-demand request after receiving a demand request and executing the demand request before completing the non-demand request.

5.   The method of claim 4 including re-queuing said non-demand request for execution after the completion of the demand request.

6.   The method of claim 1 including determining whether the cache is idle before executing a write back request.

7.   The method of claim 1 including interrupting a write back request during its execution after receiving a demand request.

8.   The method of claim 1 including executing cache flush operations when a pending write back request has been received.

9.   The method of claim 1 including executing a driver generated non-demand write back request.

10.   An article comprising a medium storing instructions that, if executed, enable a processor-based system to:

determine if there is a pending demand request to a cached disk subsystem and, if not, execute a non-demand request.

11.   The article of claim 10 further storing instructions that, if executed, enable the processor-based system to queue requests including demand requests, requests to write from the cache back to a disk drive, and requests to flush the cache.

12.   The article of claim 11 further storing instructions that, if executed, enable the processor-based system to execute said non-demand request and monitor for a demand request.

13.   The article of claim 12 further storing instructions that, if executed, enable the processor-based system to interrupt the execution of the non-demand request after receiving a demand request and execute the demand request before completing the non-demand request.

14.   The article of claim 13 further storing instructions that, if executed, enable the processor-based system to re-queue said non-demand request for execution after the completion of the demand request.

15.   The article of claim 10 further storing instructions that, if executed, enable the processor-based system to determine whether the cached disk subsystem is idle before executing a non-demand request.

16.   The article of claim 10 further storing instructions that, if executed, enable the processor-based system to interrupt the execution of a non-demand request after receiving a demand request.

17.   The article of claim 10 further storing instructions that, if executed, enable the processor-based system to execute cache flush instructions when a pending write back request has been received.

18.   A system comprising:
        a cache;
        a disk drive coupled to said cache; and
        a controller to determine if there is a pending demand request to a cached disk subsystem and, if not, implement a non-demand request.

19.   The system of claim 18, said controller to queue requests including demand requests, requests to write from the cache back to the disk drive, and requests to flush the cache.

20.   The system of claim 19, said controller to execute a non-demand request and monitor for a demand request.

21.   The system of claim 20, said controller to interrupt the execution of a non-demand request after receiving a demand request and execute the demand request before completing the non-demand request.

22.   The system of claim 21, said controller to re-queue said non-demand request after a completion of the demand request.

23.   The system of claim 18, said controller to determine whether the cached disk subsystem is idle before executing a non-demand request.

24.   The system of claim 18, said controller to interrupt the execution of a non-demand request after receiving a demand request.

25.   The system of claim 18, said controller to execute cache flush instructions when a pending write back request has been received.

EVIDENCE APPENDIX

None.

<u>**RELATED PROCEEDINGS APPENDIX**</u>

None.